Moving Average Strategy with Backtesting

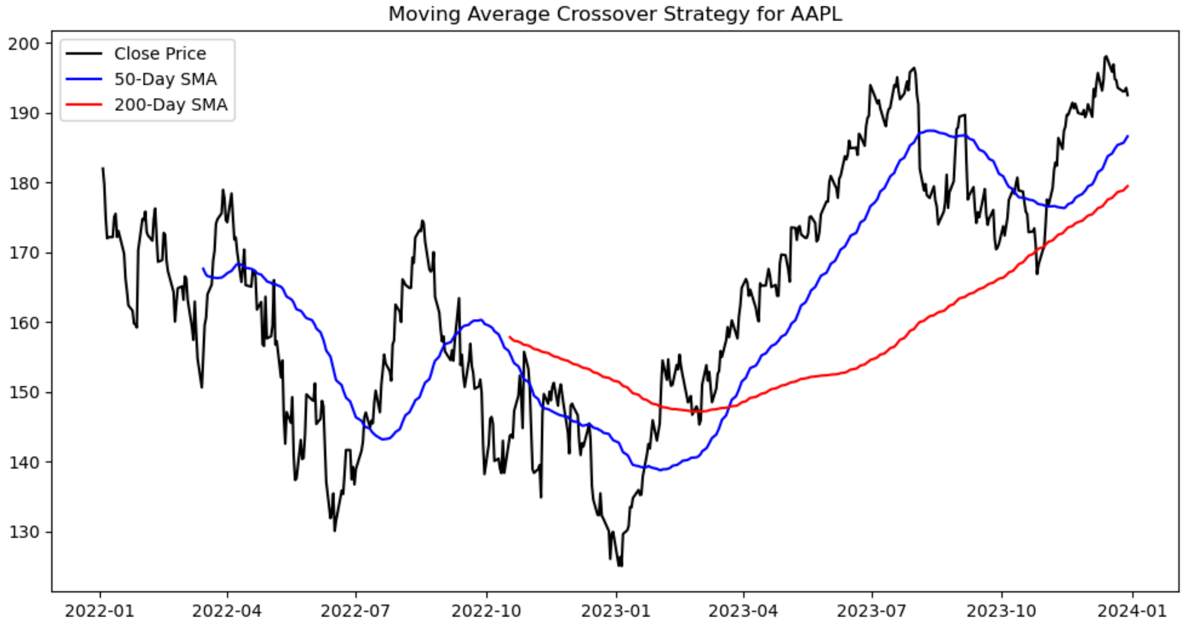
In [3]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import yfinance as yf
```

In [36]:

```
#Fetch Historical Data

ticker = "AAPL"
start_date = "2022-01-01"
end_date = "2024-01-01"

data = yf.download(ticker, start=start_date, end=end_date)
data["SMA50"] = data["Close"].rolling(window=50).mean()  # Short-term MA
data["SMA200"] = data["Close"].rolling(window=200).mean()  # Long-term MA

plt.figure(figsize=(12,6))
plt.plot(data["Close"], label="Close Price", color="black")
plt.plot(data["SMA50"], label="50-Day SMA", color="blue")
plt.plot(data["SMA200"], label="200-Day SMA", color="red")
plt.legend()
plt.title(f"Moving Average Crossover Strategy for {ticker}")
plt.show()
```

```
[*********************100%%**********************]  1 of 1 completed
```

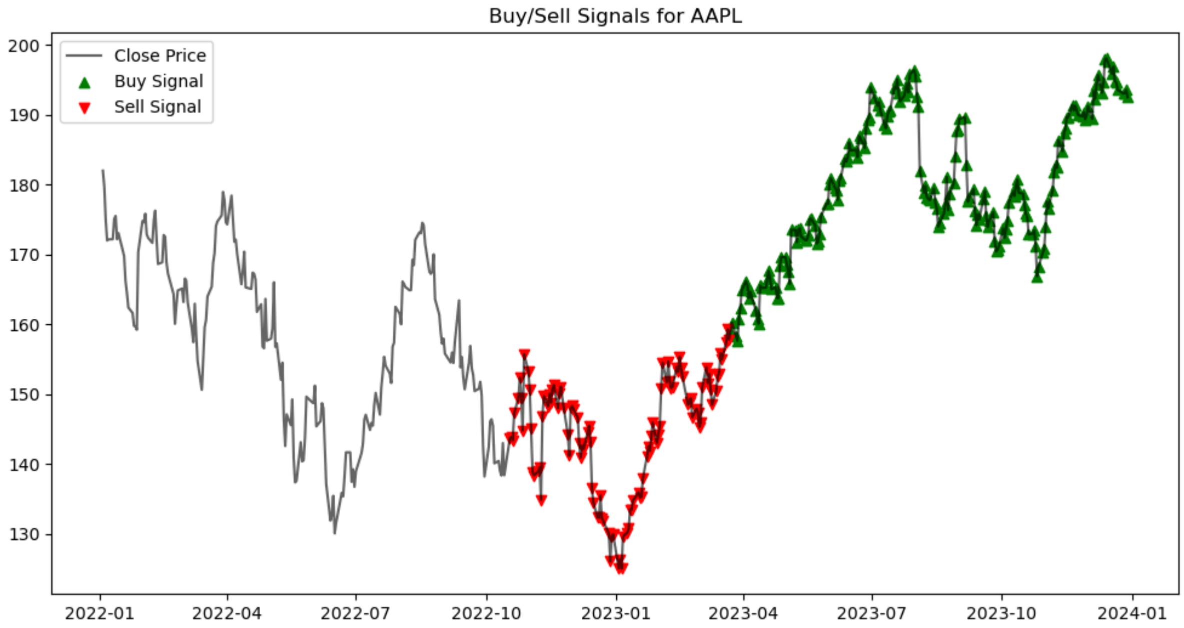


In [37]:

```
# Generate Trading signals

data["Signal"] = 0
data.loc[data["SMA50"] > data["SMA200"], "Signal"] = 1  # Buy Signal
data.loc[data["SMA50"] < data["SMA200"], "Signal"] = -1  # Sell Signal
```

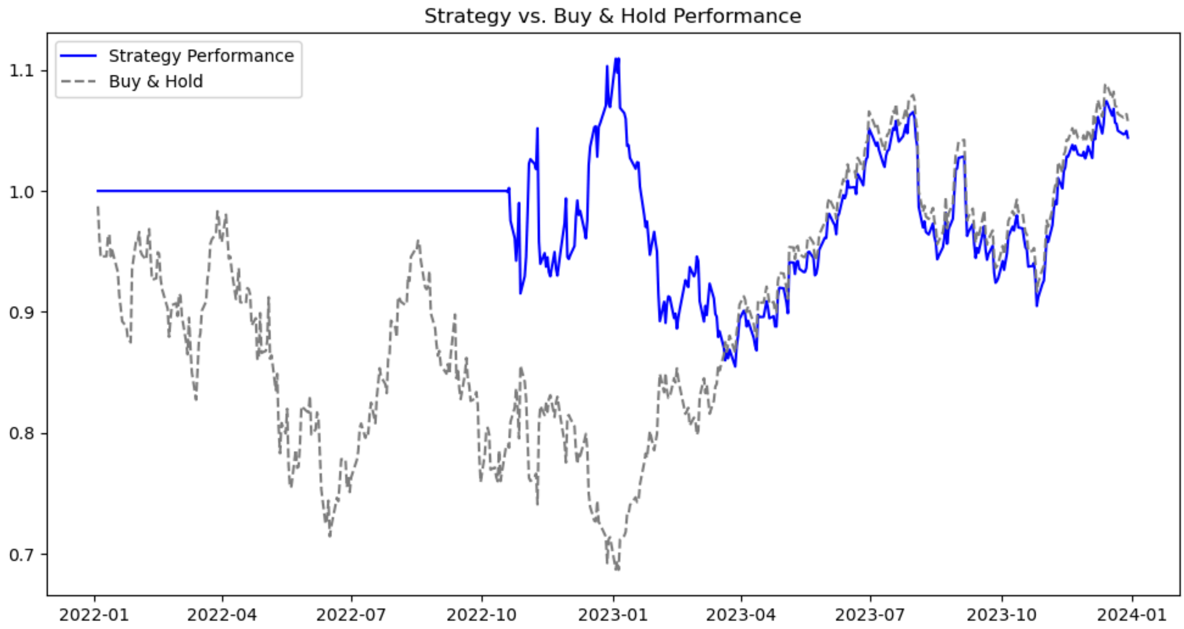
In [38]:

```
#Visualise Buy and Sell points

plt.figure(figsize=(12,6))
plt.plot(data.index, data["Close"], label="Close Price", color="black", alpha=0.6)

plt.scatter(data.loc[data["Signal"] == 1].index, data.loc[data["Signal"] == 1, "Clo
            label="Buy Signal", marker="^", color="green", alpha=1)

plt.scatter(data.loc[data["Signal"] == -1].index, data.loc[data["Signal"] == -1, "C
            label="Sell Signal", marker="v", color="red", alpha=1)

plt.legend()
plt.title(f"Buy/Sell Signals for {ticker}")
plt.show()
```



In [39]:

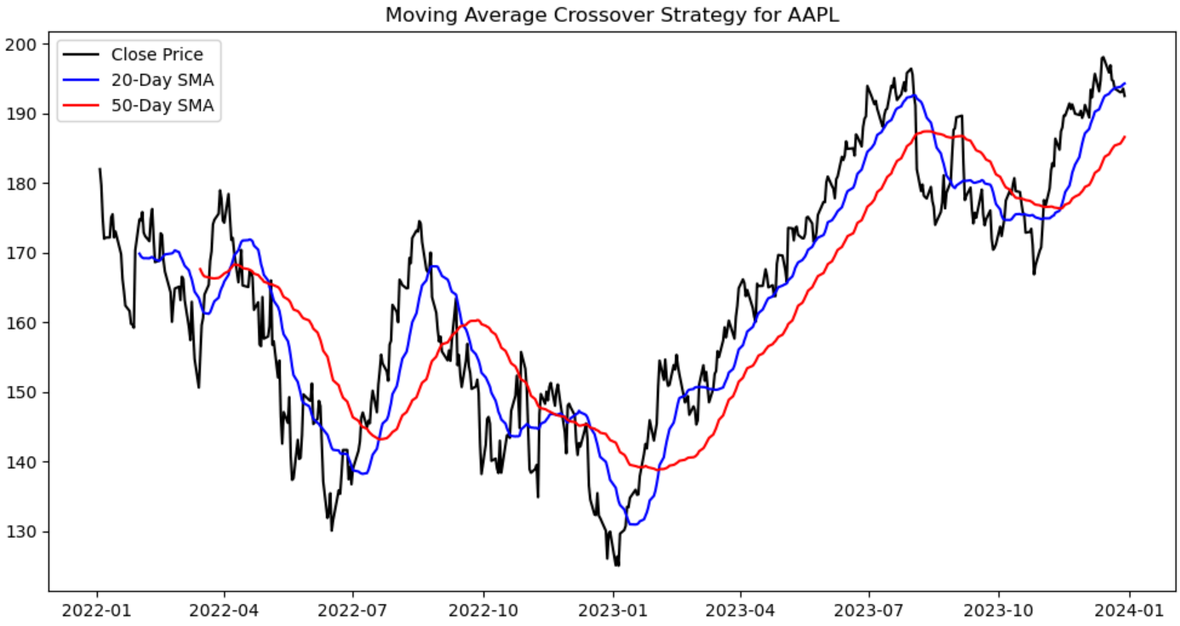
```
data["Daily Return"] = data["Close"].pct_change()
data["Strategy Return"] = data["Daily Return"] * data["Signal"].shift(1)

cumulative_returns = (1 + data["Strategy Return"]).cumprod()

plt.figure(figsize=(12,6))
plt.plot(cumulative_returns, label="Strategy Performance", color="blue")
plt.plot((1 + data["Daily Return"]).cumprod(), label="Buy & Hold", color="gray", li
plt.legend()
plt.title("Strategy vs. Buy & Hold Performance")
plt.show()
```



backtesting using 20 & 50 SMA

In [40]:

```
data["SMA20"] = data["Close"].rolling(window=20).mean()  # Short-term MA
data["SMA50"] = data["Close"].rolling(window=50).mean()  # Long-term MA

plt.figure(figsize=(12,6))
plt.plot(data["Close"], label="Close Price", color="black")
plt.plot(data["SMA20"], label="20-Day SMA", color="blue")
plt.plot(data["SMA50"], label="50-Day SMA", color="red")
plt.legend()
plt.title(f"Moving Average Crossover Strategy for {ticker}")
plt.show()
```

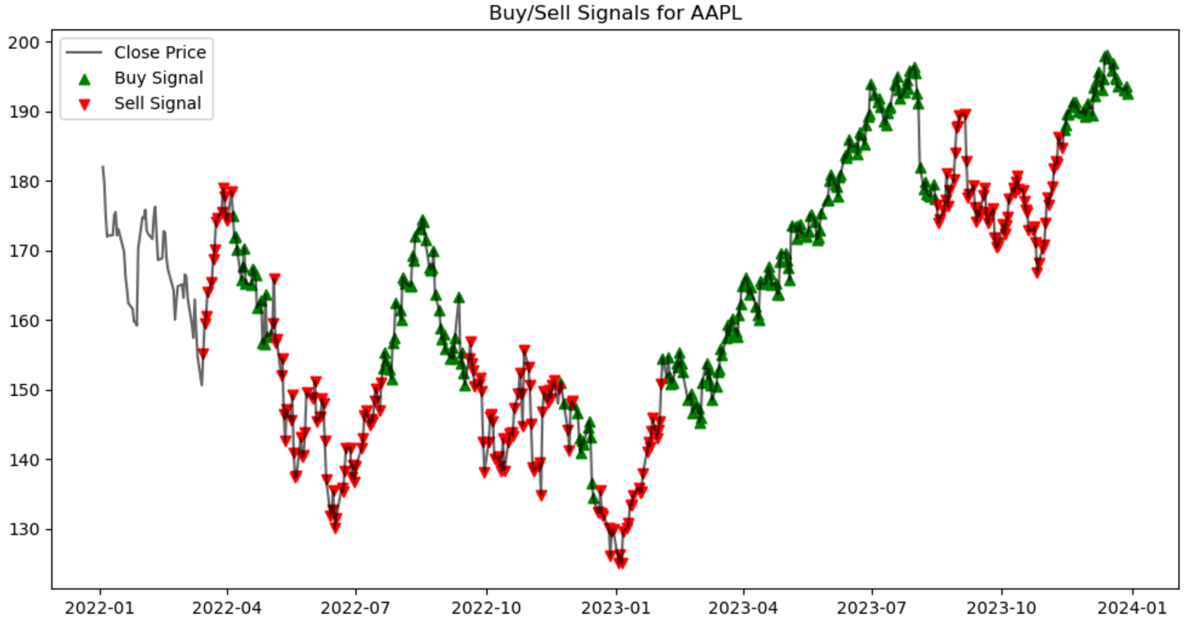


In [41]:

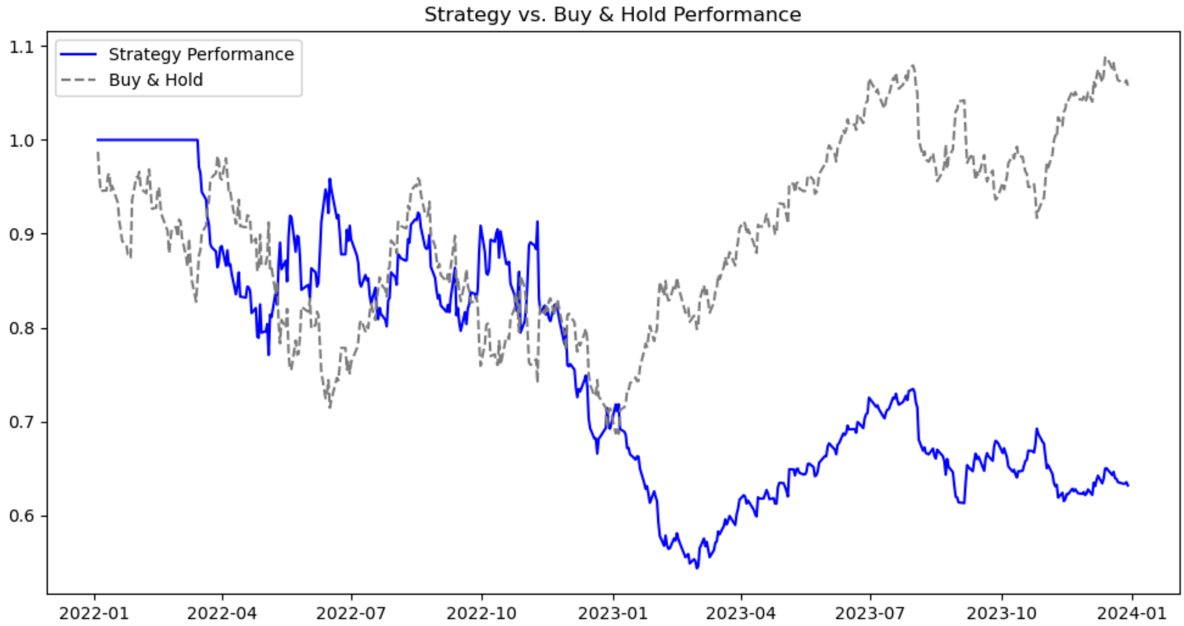
```
# Generate Trading signals

data["Signal"] = 0
data.loc[data["SMA20"] > data["SMA50"], "Signal"] = 1  # Buy Signal
data.loc[data["SMA20"] < data["SMA50"], "Signal"] = -1  # Sell Signal
```

In [42]:

```
#Visualise Buy and Sell points

plt.figure(figsize=(12,6))
plt.plot(data.index, data["Close"], label="Close Price", color="black", alpha=0.6)

plt.scatter(data.loc[data["Signal"] == 1].index, data.loc[data["Signal"] == 1, "Clo
            label="Buy Signal", marker="^", color="green", alpha=1)

plt.scatter(data.loc[data["Signal"] == -1].index, data.loc[data["Signal"] == -1, "C
            label="Sell Signal", marker="v", color="red", alpha=1)

plt.legend()
plt.title(f"Buy/Sell Signals for {ticker}")
plt.show()
```

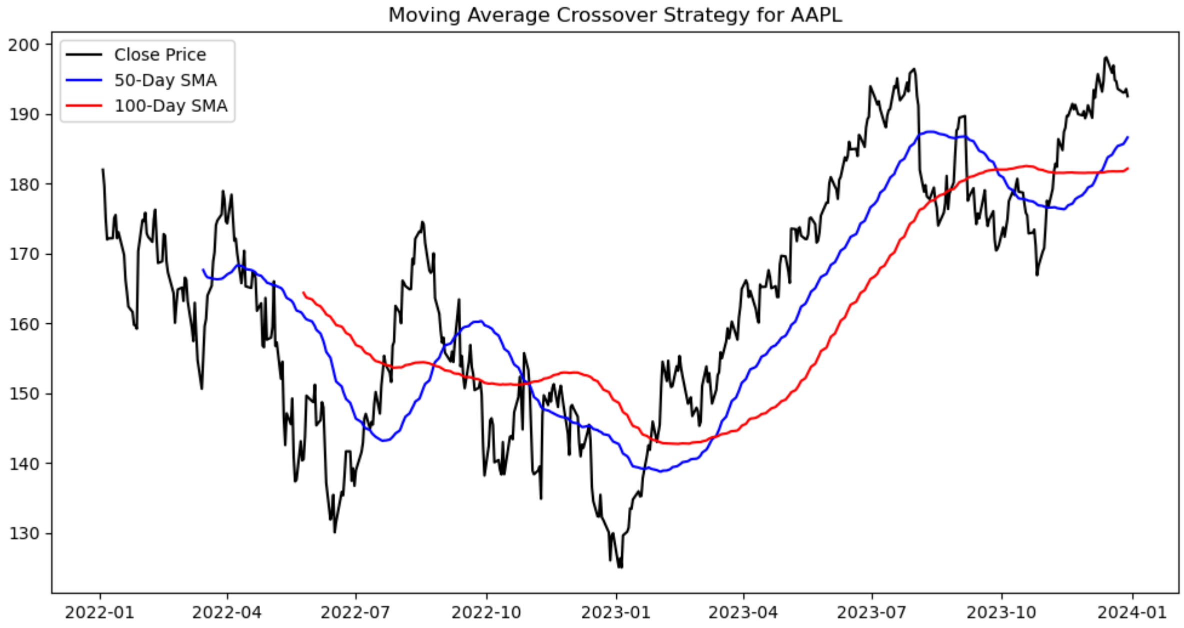


In [43]:
```
data["Daily Return"] = data["Close"].pct_change()
data["Strategy Return"] = data["Daily Return"] * data["Signal"].shift(1)

cumulative_returns = (1 + data["Strategy Return"]).cumprod()

plt.figure(figsize=(12,6))
plt.plot(cumulative_returns, label="Strategy Performance", color="blue")
plt.plot((1 + data["Daily Return"]).cumprod(), label="Buy & Hold", color="gray", li
plt.legend()
plt.title("Strategy vs. Buy & Hold Performance")
plt.show()
```



backtesting using 50 & 100 SMA

In [44]:
```
data["SMA50"] = data["Close"].rolling(window=50).mean()  # Short-term MA
data["SMA100"] = data["Close"].rolling(window=100).mean()  # Long-term MA

plt.figure(figsize=(12,6))
plt.plot(data["Close"], label="Close Price", color="black")
plt.plot(data["SMA50"], label="50-Day SMA", color="blue")
plt.plot(data["SMA100"], label="100-Day SMA", color="red")
plt.legend()
plt.title(f"Moving Average Crossover Strategy for {ticker}")
plt.show()
```

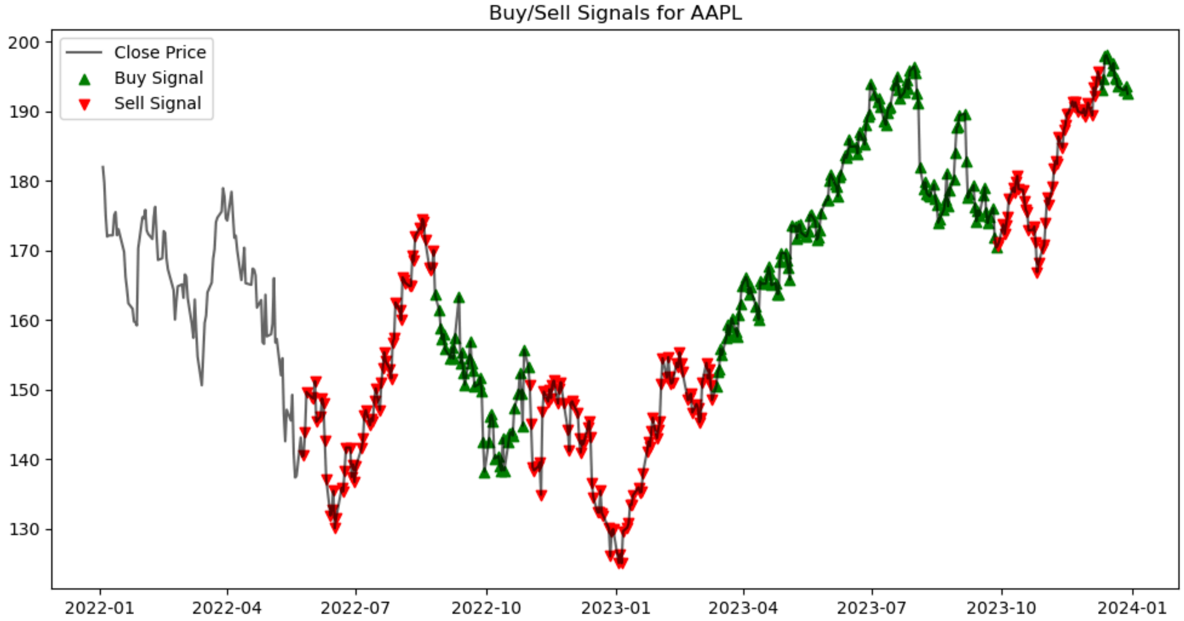


In [45]:

```
# Generate Trading signals

data["Signal"] = 0
data.loc[data["SMA50"] > data["SMA100"], "Signal"] = 1  # Buy Signal
data.loc[data["SMA50"] < data["SMA100"], "Signal"] = -1  # Sell Signal
```

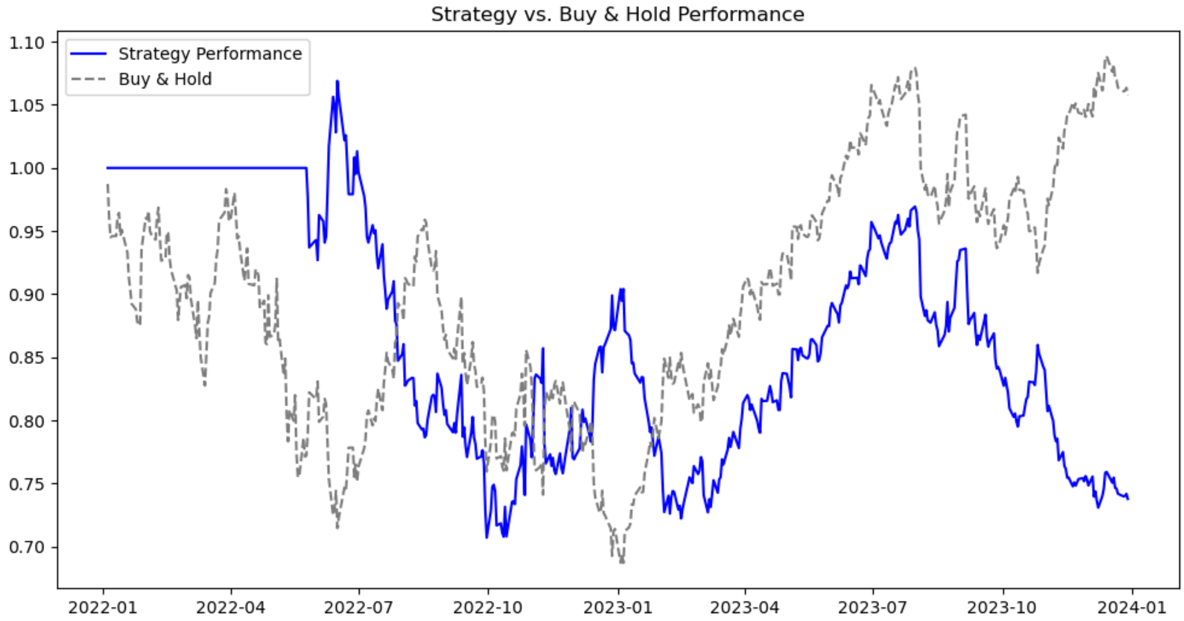
In [46]:

```
#Visualise Buy and Sell points

plt.figure(figsize=(12,6))
plt.plot(data.index, data["Close"], label="Close Price", color="black", alpha=0.6)

plt.scatter(data.loc[data["Signal"] == 1].index, data.loc[data["Signal"] == 1, "Cl
            label="Buy Signal", marker="^", color="green", alpha=1)

plt.scatter(data.loc[data["Signal"] == -1].index, data.loc[data["Signal"] == -1, "C
            label="Sell Signal", marker="v", color="red", alpha=1)

plt.legend()
plt.title(f"Buy/Sell Signals for {ticker}")
plt.show()
```



In [47]:

```
data["Daily Return"] = data["Close"].pct_change()
data["Strategy Return"] = data["Daily Return"] * data["Signal"].shift(1)

cumulative_returns = (1 + data["Strategy Return"]).cumprod()

plt.figure(figsize=(12,6))
plt.plot(cumulative_returns, label="Strategy Performance", color="blue")
plt.plot((1 + data["Daily Return"]).cumprod(), label="Buy & Hold", color="gray", li
plt.legend()
plt.title("Strategy vs. Buy & Hold Performance")
plt.show()
```



In [ ]:

In [ ]:

In [ ]: